MAIN FRAME PLUS★WARE™ NEWS
A PUBLICATION OF STSC, INC.
August 1985 Issue 2

New Release of APL★PLUS Mainframe System Highlighted In This Issue

Pre-Compiled Fast Functions Supplied With New Release

With Release 5.0 of the APL★PLUS Mainframe System you can now run complied APL functions in the VM and MVS environments. As a further convenience, STSC has provided with this new release the workspace 3 FASTFNS, consisting of some pre-compiled functions to get you started.

Most of the functions provided with the FASTFNS workspace are far faster than other APL functions that do the same job. Some are compiled APL functions, while others are written in Assembler language or our own systems programming language called TABL (for Tool and Application Building Language). Most of the functions in the FASTFNS workspace are at least twice as fast as standard APL alternatives, while some are nearly 2000 times as fast.

There are presently 26 functions in the FASTFNS workspace. They are helpful for checking input, formatting output, text processing, sorting, ranking, counting, table lookup, communications, and converting data from non-APL sources.

One of the most useful functions in the FASTFNS workspace is *MATIOTA*, a function that looks up rows of a character matrix in another character matrix. For matrices with 300 rows, this is 78 to 146 times as fast as the commonly used method based on the ∧ . = inner product. The speed difference is so large because inner product compares every row with every row, while *MATIOTA* uses hashing to classify the rows into groups that might match, then compares only within the groups. As a result, each row of the right argument usually is compared with one or two rows of the left argument that fall in the same class.

Mark Baur-Peters, an STSC employee, used *MATIOTA* together with another of the FASTFNS to solve a related problem for a customer. The customer's

continued on page 4

New Release Provides Monitor Function For Finding CPU Bottlenecks

With Release 5.0 of the APL★PLUS Mainframe System, a new system function, □*MF* has been provided. The □*MF* monitor function provides an easy way to find where CPU time is spent in an application. Finding the slowest functions in a workspace (the CPU "bottlenecks") allows you to concentrate speedup efforts on those areas that can make the most difference.

Workspace 3 MFFNS, also provided with the new release, contains utility functions which use □*MF* to produce formatted reports summarizing CPU usage. Two main functions in the *MFFNS* workspace are *CUMSUM* and *LTIMES*. *CUMSUM* produces a table showing the distribution of CPU time across all the functions in an application. For example, if we had a collection of programs which implements Rubik's cube, the function *CUMSUM* can be used to see where CPU time was being spent within the application.

```
      1 □MF □NL 3          ⍝ TURN MONITORING ON
      PLAY                 ⍝ RUN THE APPLICATION
      )COPY 3 MFFNS GRPCUMSUM
      CUMSUM □NL 3         ⍝ RUN CUMSUM

WS:  4705386 RUBIK        JUN 17, 1985   4:28 PM

              CPU          CUM     NUM    MAX   LINES
FUNCTION     SECS    PCT   PCT   CALLS   ITER  IN FN
--------    -----   ----   ---   -----   ----  -----
AFTER       .2128   85.9    86       3    121     13
SHOW        .0181    7.3    93       4      4      7
PLAY        .0168    6.8   100       1      4     13

TOTAL       .2477
```

The table above shows that about 86% of the time was spent in the *AFTER* function.

Knowing where CPU time is spent in an application is essential to being able to speed up the application. For example, speeding up *PLAY* in the previous example by

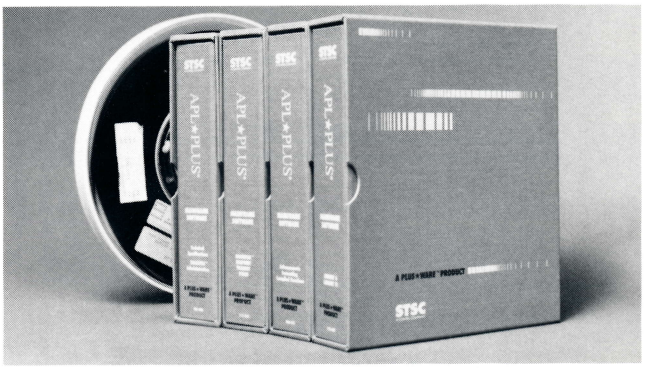
continued on page 4

All New Documentation Available With New Release

Enhanced and repackaged documentation is now available with Release 5.0 of the APL★PLUS System Mainframe Software Product. Printed in the popular and convenient small loose-leaf binders, the new documentation is easier than ever to use at a terminal or microcomputer.

The 16 manuals that document the mainframe APL product are now contained in 4 small binders, as follows:

- The APL★PLUS Enhancements Binder contains three manuals: *Enhancements to VS APL,* which describes in detail the use of each system function, variable, command, and APL language extension; the *Formatting User's Guide,* which describes the powerful report formatting features available with □*FMT*; and the *FASTFNS User's Guide,* which describes the new set of pre-compiled utility functions that are available as a part of Release 5.0.
- The APL★PLUS SHAREFILE™ System Binder contains five manuals: *SHAREFILE System User's Guide,* which shows you how to use STSC's component-based, shared access file system; *Using Shared Files With Non-APL Programs,* which shows you how to use the file system with non-APL programs; *FCGEN User's Guide,* which shows you how to convert files to or from an external dataset; *FILESORT Utility Manual,* which contains instructions for sorting large data files; and *FILEPRINT Facility User's Guide* for printing files on a high-speed printing facility.
- The APL★PLUS Programmer Tools Binder contains clear and detailed descriptions of five workspaces that contain valuable programming tools. The workspaces include 6 SORT and 6 INPUT for implementing sorting and inputing routines in your applications; 6 PARTFNS for developing sophisticated Boolean functions and techniques; 11 FNED for simplified function editing; and 11 WSDOC to aid in documenting workspaces.
- The APL★PLUS Administrator's Binder contains step-by-step instructions for installing, maintaining, and administering the APL★PLUS Mainframe System. The binder includes three manuals that provide technical specifications for the system, instructions for administering the SHAREFILE System, and complete installation instructions.



We think you will like the organization and clarity of this stylish and convenient new documentation package. Additional copies of any of the four binders are available for $20 each from your STSC representative.

Improved Tape Handling Routines Supplied With New Release

Extensions to the tape mounting procedures used for SHAREFILE System backup under VM/SP have been provided with Release 5.0 of the APL★PLUS Mainframe Software System. The extensions provide an interface routine that allows a customer-written EXEC to control tape mounting as an alternative to the current procedures.

The SHAREFILE utility programs can still control tape mounting directly. Tape mounting is requested by messages to the operator, and any tape mounting faults are reported to the operator for the appropriate action. However, this procedure does not allow you to interact with any other tape management system.

Now an alternative method is provided as well. Release 5.0 contains a new interface routine that lets you tailor tape mounting procedures to your specialized requirements. STSC provides a simple interface routine, called SFSTAPEX, which relays pertinent tape mounting information from SHAREFILE System utility programs to an EXEC that you write. SFSTAPEX passes information like program name, the type of utility operation, tape drive virtual address, tape volume serial number, and more to your EXEC. Your EXEC includes your special instructions for job initializing, tape mounting and demounting, job termination, and whether to try again or terminate in the event of failure, based on your particular needs. Your EXEC can also include any special instructions required by a particular tape management system like VM Tape. Tape input and output is still performed directly by the SHAREFILE System Utility Programs.

continued at top of page 3

MAIN FRAME PLUS★WARE™ NEWS
A PUBLICATION OF STSC, INC.

Mainframe PLUS★WARE News is published by STSC, Inc. and is distributed free of charge to its customers. Comments regarding the newsletter should be sent to:

Mainframe Newsletter Editor
STSC, Inc.
2115 East Jefferson St.
Rockville, MD 20852

© 1985 STSC, Inc. All rights reserved. Printed in U.S.A.

APL★PLUS is a registered trademark of STSC, Inc. PLUS★WARE and SHAREFILE are trademarks of STSC, Inc.

A sample EXEC is provided with the new release. This simple but worthwhile example is intended to guide you in setting up your own working EXEC, together with a procedure for activating them when you are ready.

Further details and a full description of the interface routines is included in an 8-page supplement to the Release 5.0 Installation Guide provided on your tape. The file is called SFSTAPEX LISTING. Should you have any questions or need to discuss some aspect of the new tape mounting mechanism after reading the supplement, call STSC's Mainframe Software Product Support Team at 301-984-5160 for assistance.

Meet The Support Team

In this issue of Mainframe PLUS★WARE News, we'll introduce you to STSC's Mainframe Software Product Support Team (SPST).

STSC developers work continuously to enhance the APL★PLUS Mainframe Product, hence the sequence of new releases. Then with each release, SPST "packages" the product and works at making the product overall more robust. Thereafter, they help install the product, advise customers on adapting the product to their own special environment, and analyze and work to correct problems they've failed to anticipate. Occasionally, SPST is asked to deal with system changes caused by external disturbances—a new release of the operating system or a disk hardware failure with no recent backup data, for example. Thus, SPST brings to the Mainframe Software product a system programming service specializing in APL. And in the process, working with and inside APL, they find fascinating and worthwhile work.

Carol Simmons, Manager of the Team says "There is nothing more exciting than resolving a problem that was totally baffling 10 minutes earlier. It is a real challenge solving problems remotely when I don't have access to the system I'm fixing and I can't reproduce the problem locally. After a while, I learn to ask the correct questions and never assume anything."

Chris Lee, a systems programmer on the support team, echos Carol's statement when he says: "Product support offers me the best of two worlds—personal creativity coupled with the satisfaction of helping others. It is exciting to be involved with the development of new products, especially when we are able to see those products put into use. It's also gratifying to interact directly with our customers to help them get the greatest benefit from our work."

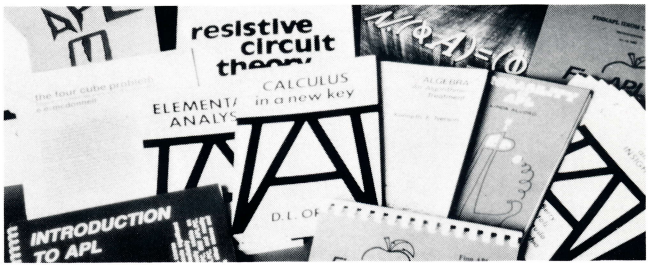
The Mainframe Software Product Support Team (left to right): Carol Simmons, Chris Lee, and Tom Pritchard.

Among the many things he had to say about supporting the APL★PLUS System, Tom Pritchard, Systems Programmer, says "I think we're blessed with a great job. We work with APL and inside APL, both of which are fascinating and useful pursuits. Extensive problem-solving in a wonderful environment is amply challenging and rewarding. In the process, we invariably enjoy a super rapport with the customer's system programming staff."

STSC Acquires APL Press

STSC and I. P. Sharp Associates Limited of Toronto, Canada, announced STSC's acquisition of APL Press of Palo Alto California. APL Press publishes and distributes reference books, tutorials, and self-study courses dealing with the APL programming language. APL Press publications include several introductions to APL that focus on the use of APL in multiple disciplines, such as engineering, and that teach various topics, such as algebraic calculus. Two current APL Press publications combine software with manuals.

According to John W. Myrna, senior vice president in charge of STSC's APL Product and Services Division, "We're excited by the opportunity to combine the book publishing experience of APL Press, especially in the APL training area, with our software product publishing experience in order to offer innovative packages such as self-contained APL courses."



Included with this issue of Mainframe PLUS★WARE News is a list and brief description of each APL Press publication and tutorial. An order form is provided on the back of the list for your convenience. For additional copies of the order form, contact your STSC representative.

Compiled Functions *continued from page 1*

application needed to look up rows of one integer matrix in another integer matrix. The fast function *CHTYPE* (the name is an abbreviation for CHange TYPE) was first used to convert the integer matrices into character matrices. That is, each integer was converted into the four characters that have the same binary representation. A statement like *CHMAT ← 'CI' CHTYPE INTMAT* was used to perform the conversion. The left argument for *CHTYPE* contains one-letter abbreviations for the result and argument types. In this case, the conversion is to character type from integer type.

Mark reported that even including the time to convert the data to characters, this solution was 100 times faster for typical data (950 rows and 3 columns for the left argument and 200 rows and 3 columns for the right argument) than the previous method. Another solution that was considered was to pack the data for each row as a single integer, and use dyadic iota to do the searches. That approach would have conflicted with plans to increase the range of values in the rows, and besides, the solution based on the FASTFNS was even faster.

Release 5.0 gives you the ability to run compiled functions. To produce compiled functions you need the APL★PLUS Compiler Product. The Compiler is now begin tested at select customer sites and will not be offered as a commercial product until the test period is nearly over. Although the Compiler is not yet available for use on your own computer, functions can be compiled on STSC's Remote Computing Service and can be run on your machine. This approach can be worthwhile in cases where a large amount of CPU time is used in a relatively small part of an application. (See the article "New Release of Mainframe System Provides Monitor Function for Finding CPU Bottlenecks," in this issue.)

Documentation for the new 3 FASTFNS workspace is included in the new Enhancements Manual. If you would like more information on compiling functions or on finding CPU bottlenecks in your applications, contact Clark Wiedmann or Jim Weigang of STSC's Haydenville, Massachusetts, office at 413-268-7500.

Monitor Function *continued from page 1*

a factor of ten would have considerably less impact than speeding up *AFTER* by the same factor.

Once the slowest functions have been identified, the *LTIMES* function can be used to determine where time is being spent within individual functions.

```
      )COPY 3 MFFNS GRPLTIMES
      LTIMES 'PLAY'

  ALL  HERE  ITER        ∇PLAY;S
    0     0     0  [1]    ⍝IMPLEMENTS RUBIK'S CUBE
    6     6     1  [2]    ⍎(0=⎕NC'CUBE')/'CUBE←NEWCUBE'
   64    11     1  [3]    ⎕←SHOW CUBE
   20    20     4  [4]  L1:⍞←S←'COMMAND?  '
   28    28     4  [5]    S←(⍴S)↓⍞
   20    20     4  [6]    ⍎(','=1↑S)/'S←⍎S'
   14    14     4  [7]    ⍎('*'=1↑S)/'CUBE←NEWCUBE'
   11    11     4  [8]    →(0=⍴S)/0
   16    16     3  [9]    S←(S∈'LCRBHFUMDLCRBHFUMD')/S
 2465     5     3  [10]   CUBE←CUBE AFTER S
  178    33     3  [11]   ⎕←SHOW CUBE
    3     3     3  [12]   →L1
                         ∇
 2844   168     1
```

The second column labeled "HERE" gives the CPU time spent directly on each line of the function; the first column labeled "ALL" includes in addition time spent in subroutines. The "ITER" column gives the number of times each line was executed. The CPU bottleneck in the *AFTER* function shows up clearly in this display.

The ⎕*MF* system function is documented in the Enhancements Manual; *CUMSUM* and *LTIMES* are documented in the *MFFNS* workspace.

MAIN FRAME PLUS★WARE™ NEWS

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852

Return Postage Guaranteed

BULK RATE
U.S. POSTAGE
PAID
PERMIT No. 4590
ROCKVILLE, MD 20850